# Newsletter

## October 1991

*In this issue...*

- **Amiga Developers Conference**
- **Timebase Correctors Revealed**
- **A2320 Display Enhancer**
- **Questionnaire Results**

# HONORARY CUGI OFFICIALS

| | |
|---|---|
| Chairman | Geoffrey J. Reeves |
| Treasurer | Brian Ward |
| Membership Secretary | Tom Kinsella |
| Chief Librarian | Leon Hurst |
| Amiga Librarian | Rocco Matassa |
| PC Librarian | Brian Ward |
| C64/128 Librarian | Geoffrey J. Reeves |
| Communications | Declan McArdle |
| Newsletter Editor | Eddy Carroll |
| Discounts Officer | Geoffrey J. Reeves |

CUGI meeetings are held every second Friday, at 8:00 PM in the computer room at St. Andrew's College, Booterstown. All correspondence should be addressed to:

**CUGI**
c/o St. Andrew's College,
Booterstown,
Blackrock,
Co. Dublin.

All articles in this newsletter are copyrighted by the author unless otherwise stated. If you wish to reprint an article, write to its author c/o CUGI at the above address.

# Newsletter
## *Contents*

**Volume 3 Number 4**
**October 1991**

| | | |
|---|---|---|
| Editorial | Eddy Carroll | 2 |
| Questionnaire Results | Geoffrey J. Reeves | 4 |
| Developers Conference Notes | Commodore U.S. | 9 |
| Devcon Keynote Address | Jeff Scherb | 11 |
| Timebase Correctors Revealed | Colm O'Rourke | 15 |
| A2320 Display Enhancer | Karl Jeacle | 17 |
| The Hacker | Brian Ward | 20 |
| Graphics Events, Part 2 | John Pritchard | 26 |
| Professional Draw V2.0 | Colm O'Rourke | 29 |
| Startup Sequences | Jamie Ruane | 31 |
| For Your Diary | Secretary | 35 |
| Data Structures in C, Part 2 | Kevin McDonnell | 36 |

# Editorial

Welcome to the autumn edition of the newsletter. This issue is the first to be produced exclusively using AmigaTEX—no more hack jobs with *Professional Page* to get the front cover printed.

This startling advance in typesetting technology is a result of my obtaining the latest version of AmigaTEX a few months ago. The new version (V3.1b) includes full support for PostScript graphics, along with some other nice features not present in the original. A special mention must go to Paul Ockenden of Industrial Might & Logic (AmigaTEX's UK distributors) for service beyond the call of duty. Less than two weeks ago, I suddenly remembered I hadn't upgraded the laser printer driver for TEX, and at almost the same time, An Post decided to go on strike. Paul kindly sent me the driver via modem, ensuring that the missive you are reading could be printed.

A lot of you have been asking for more artwork in the newsletter. Hopefully, you'll see some evidence of that in this issue. Now that PostScript is available, a whole range of possibilities are available for jazzing things up. If anybody out there would like to submit some structured drawings (in either Professional Draw format or raw PostScript), they would be warmly welcomed.

A fair amount has happened since July. There have been two Amiga developers conferences hosted by Commodore, one in Denver and one in Milan. I've included some press releases from the Denver Devcon in this issue. I paid a visit to the Milan conference, along with about 150 other Amiga programmers from around Europe. Overall, the conference was excellent. There were a lot of interesting sessions to attend, plus a large (and I mean *large*) set of conference notes to take home.

Much of the conference was devoted to Kickstart 2.0. CATS US has been working on several products for developers and users to enhance the system. One of these is a standard installation program to make it easy to install third-party applications onto a hard disk. I saw this running, and it looked really nice. Although it sports the Workbench 2.0 look, it also runs under Workbench 1.3. It can be fully programmed via a script language, and has a variety of nice features which make it far superior to the usual AmigaDOS-script style installation method most packages use.

Also shown were AppShell and AppBuilder. AppShell provides a framework around which a full application can be built—it takes care of all the messy business of managing menus, gadgets, ARexx, and so on, leaving you free to concentrate on the functionality of your code. AppBuilder works in conjunction with AppShell to let you design your interface using the mouse rather than by writing code. Although not yet finished, what I saw looked fairly impressive.

There was also a lot of CDTV stuff at the show. While many of you were probably not overwhelmed by the CDTV demo last June (myself included), rest assured that there is good software on the way. Some of the discs I saw there that were particularly good included Trivial Pursuit, Music Maker, and various demos of CDXL, Commodore's full motion video technology (not quite full motion, but good enough to be useful). The advanced video adapter was also demonstrated, extending the CDTV's colour range to 4 million colours. This item is particularly interesting because it looks like it may well work on standard Amiga's also, and it is likely to be very low cost.

By now, the first batch of Amiga 500 Plus's should be reaching the shops. The Amiga 500 Plus is the A500 you all know and love, but with the extended graphics chip set (the infamous *Productivity Mode*), 1 meg of CHIP ram, a built-in battery-backed clock, and Kickstart 2.04 in ROM. The trapdoor memory expansion slot underneath has been enhanced to allow an additional 1 meg of CHIP ram to be plugged in (for a total of 2 megs), and the usual 8 megs of FAST ram can still be plugged into the side expansion port.

Late news just in from the US—Newtek have just raised the price of the Video Toaster from $1,595 to $2,495. The new package comes with version 2.0 of the Toaster software (supposedly much enhanced). The new software is also available as an upgrade to existing owners for $300. This news won't have much immediate effect on people this side of the pond, but it does mean that if a PAL Toaster ever sees the light of day, it could be significantly more expensive than had been hoped.

That's about it for now. Please get your articles for the next issue in as soon as possible, preferably before Christmas, by 3 January at the latest. Until next time...

E.C.

# Questionnaire Results

*by Geoffrey J. Reeves*

Our last Newsletter was issued along with a questionnaire and many of you took the time and trouble to fill it in and send us the results. I have compiled the results into a variety of tables and diagrams which I hope will be of interest.

From a membership of nearly 70, I received 22 filled-in questionnaires so what follows represents about 33% of the group. To be honest, I thought I'd have received more but I presume the others were very happy with the way things are. Let me begin with the breakdown of ages and membership in general.

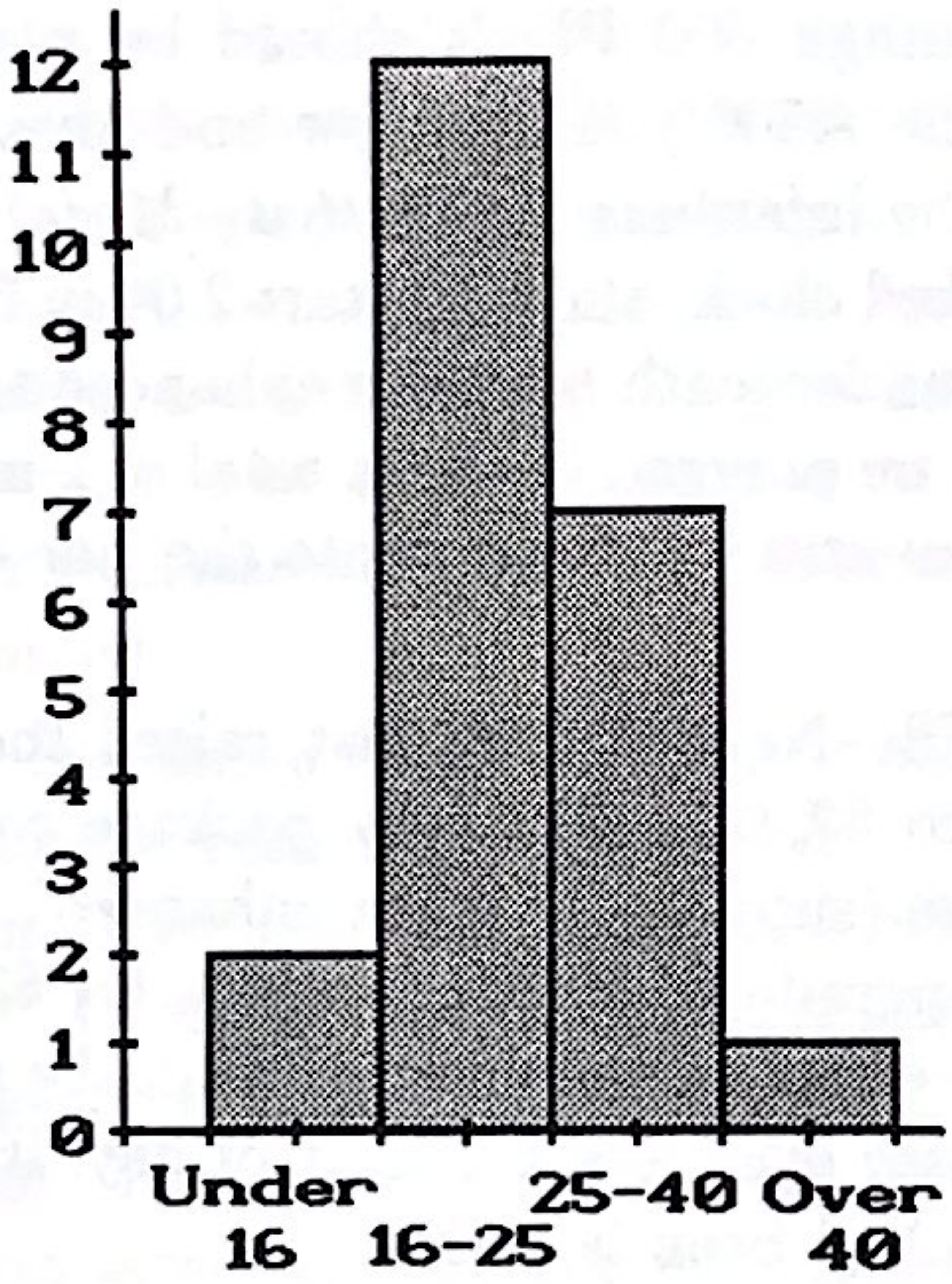


*Figure 1: Age groups*

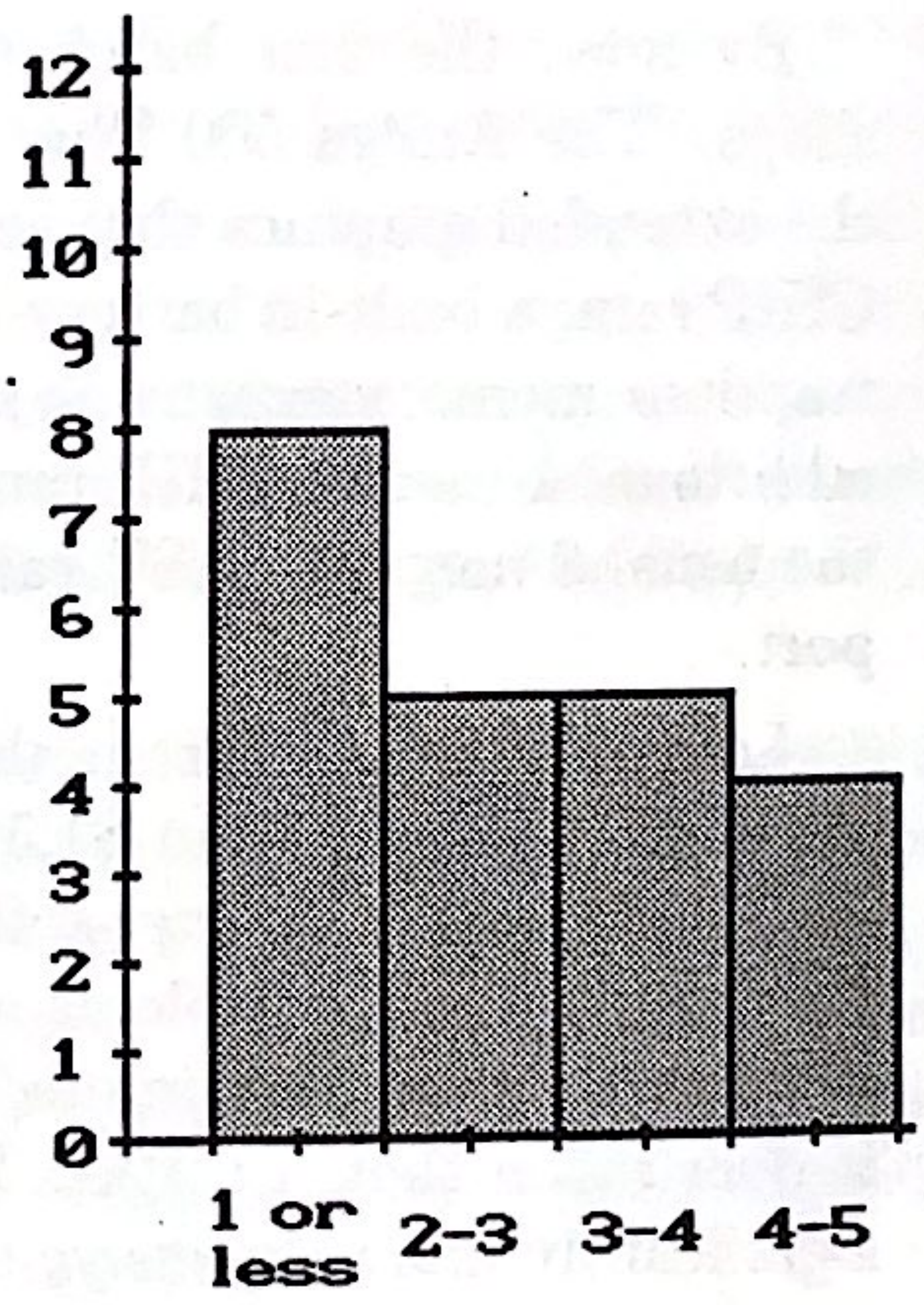


*Figure 2: Years as a member*

Figure 1 shows the number of each age. If you feel the need to get percentages then $12/22 \times 100\% = 55\,1/2\%$ of those who replied were in the 16–25 age band. No surprises there, but the length of memberships (figure 2) shows that many of our members are new. I hope we, as a group, have met their expectations—I reckon it's more likely that they won't speak up.

The wide range of computers owned proves that we are *not* just an 'Amiga club'. CUGI will try to support all Commodore computers—just look at the range in figure 3!

As for the uses to which these are put, I was delighted to see such a wide range as shown in figure 4. I must point out that *Comms* wasn't included in

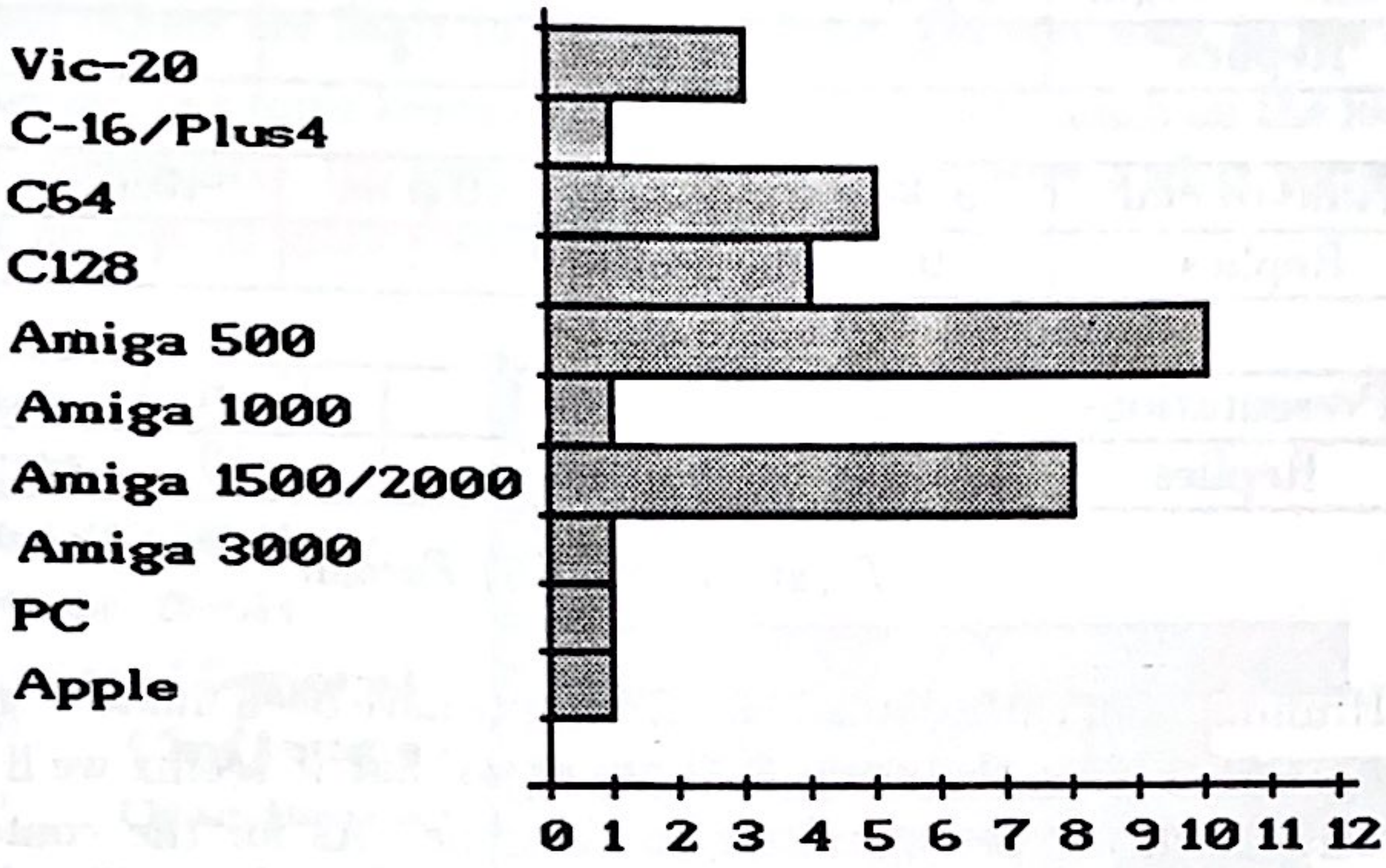


*Figure 3: Computers owned by the members*

the original questionnaire (my fault—my under-used modem is a haunting reminder of the omission). The score it received was based on members adding it to the list. I wonder did *DTP* and *Sound/Music* receive the same treatment...

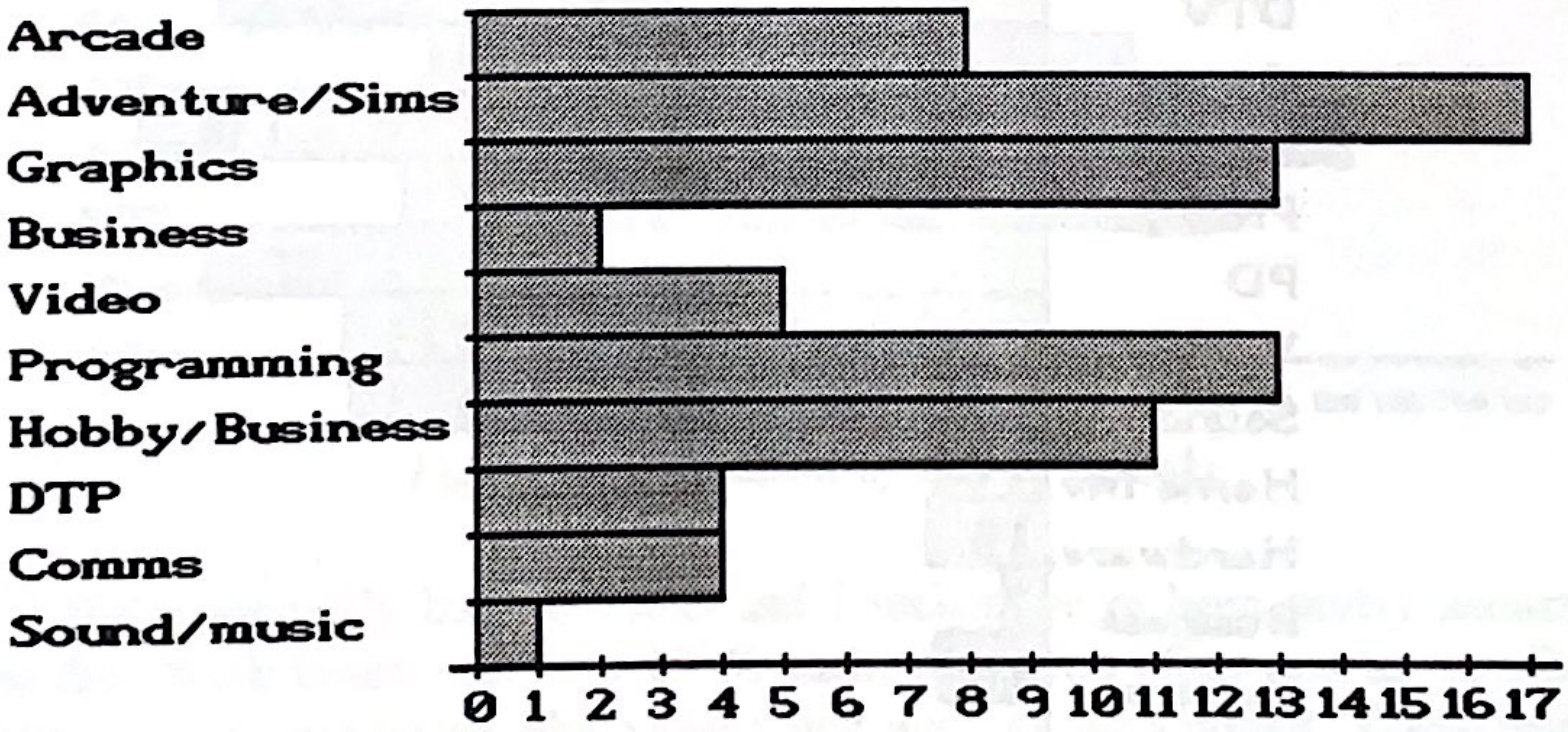


*Figure 4: Members Computer Usage*

You were then asked about the CUGI meeetings—one member lived 110 miles away (many thanks, John, for replying!), one more suggested meeting every three weeks and the remaining 20 were happy with the fortnightly arrangement.

It seems that you want fewer main items... but then, how long do you want the meetings to last? Let's see:

| Time to begin | 8 p.m. | 8.15 | 8.30 | 8.45 | 8.00–8.30 |
|---|---|---|---|---|---|
| Replies | 5 | 9 | 4 | 1 | 2 |

| Time to end | 9.30 | 9.45 | 10 p.m. | 10.15 | 10.30 |
|---|---|---|---|---|---|
| Replies | 0 | 1 | 5 | 10 | 5 |

| Presentations | 1–2 | 3–4 | Up to 5 | Over 5 |
|---|---|---|---|---|
| Replies | 0 | 6 | 10 | 4 |

*Figure 5: Meeting Format*

Hmmm... an earlier start, eh? Attempts have been made to get meetings going earlier than the usual 8.30 pm or so, but it seems we'll have to do better! Back to the committee on that one. As for the content, that is up to the *members*—committee members do present many of the items but they really appreciate others doing a presentation or two! Recalling that you awarded 0 to 10 for each topic, here's what you thought:

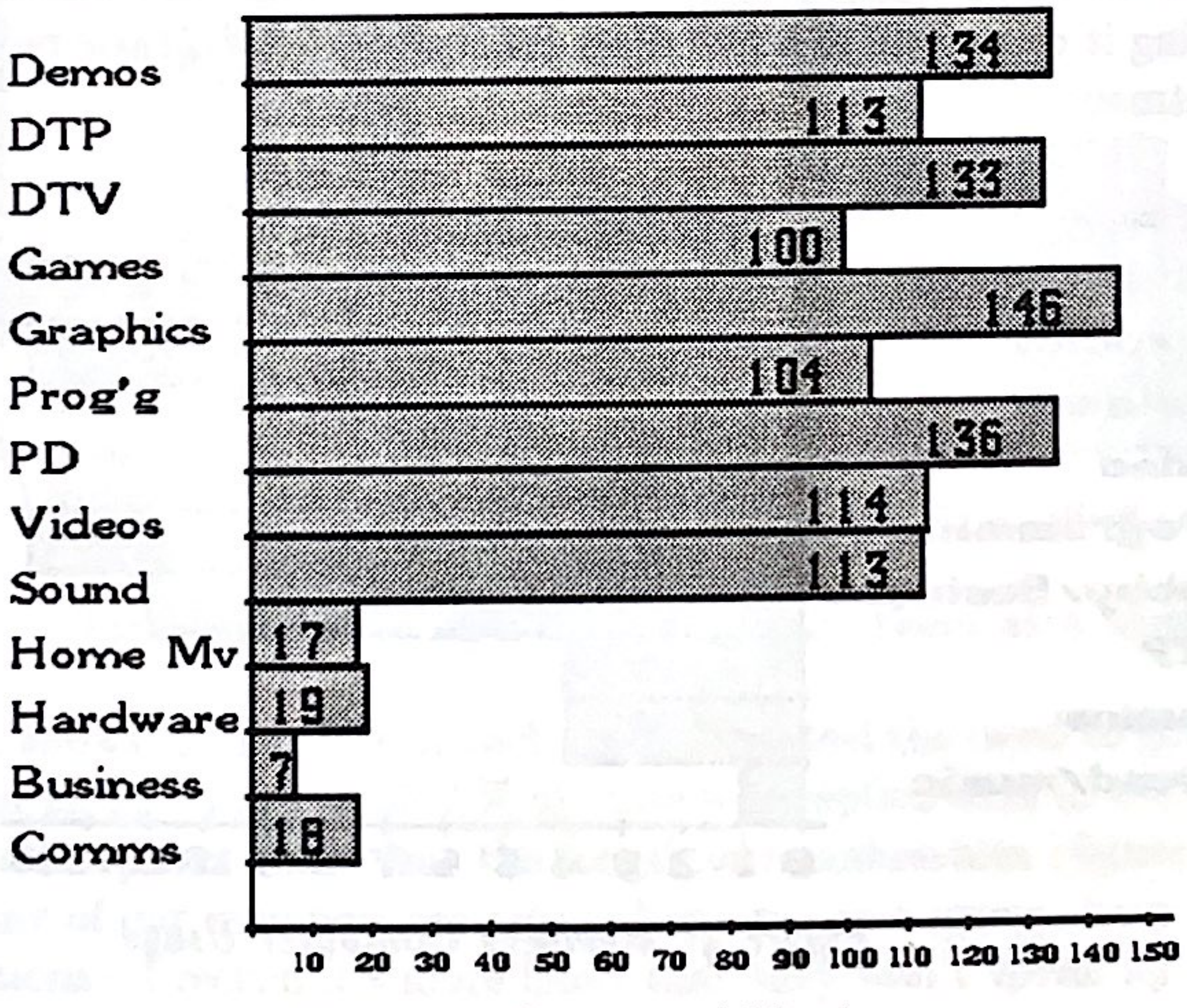


*Figure 6: Content of Meetings*

I'm not sure what you expect of *Home Movies* but I again expect that *Hardware reviews* received a low score because it wasn't on the original list (my fault... again). I reckon that many of our hardware demos have been very much appreciated since they can act as a warning as much as an encouragement. Keep us informed—demos of a flicker fixer, an accelerator card and others are likely in the near future. Do you want to see these?

However, one topic keeps us, the committee, very much on the lookout for helpful comments—the spending of *your* money! Here's what you suggested (based on a 0–10 marks per item):

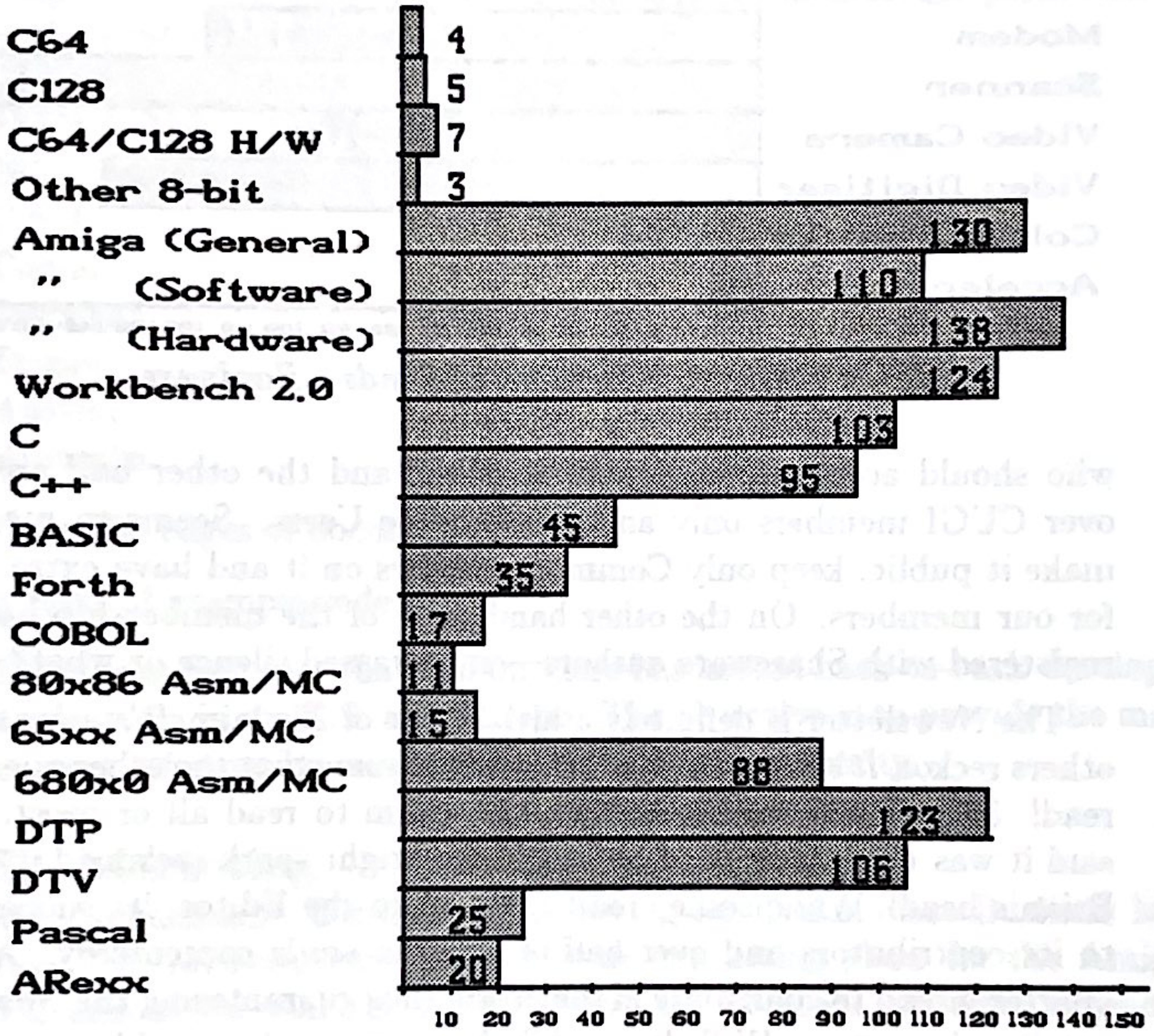


*Figure 7: Allocation of Funds—Books*

We're regularly buying books and I reckon we've been pretty accurate so far. With books costing £10–35 each, the group offers you an excellent chance to access books which you might not buy for yourself. While books on languages such as C really need to be owned rather than borrowed, the same cannot necessarily be said for the expensive ROM Kernel Manuals or books on graphics. The possibilty of buying hardware seems to have caught your imagination even more. Reminding you that we only ask for a £15 annual subscription, figure 8 shows what you'd like to see.

For the future, 80% of you think that setting up a Bulletin Board would be a good idea (even if you might not use it). You seem to be split over

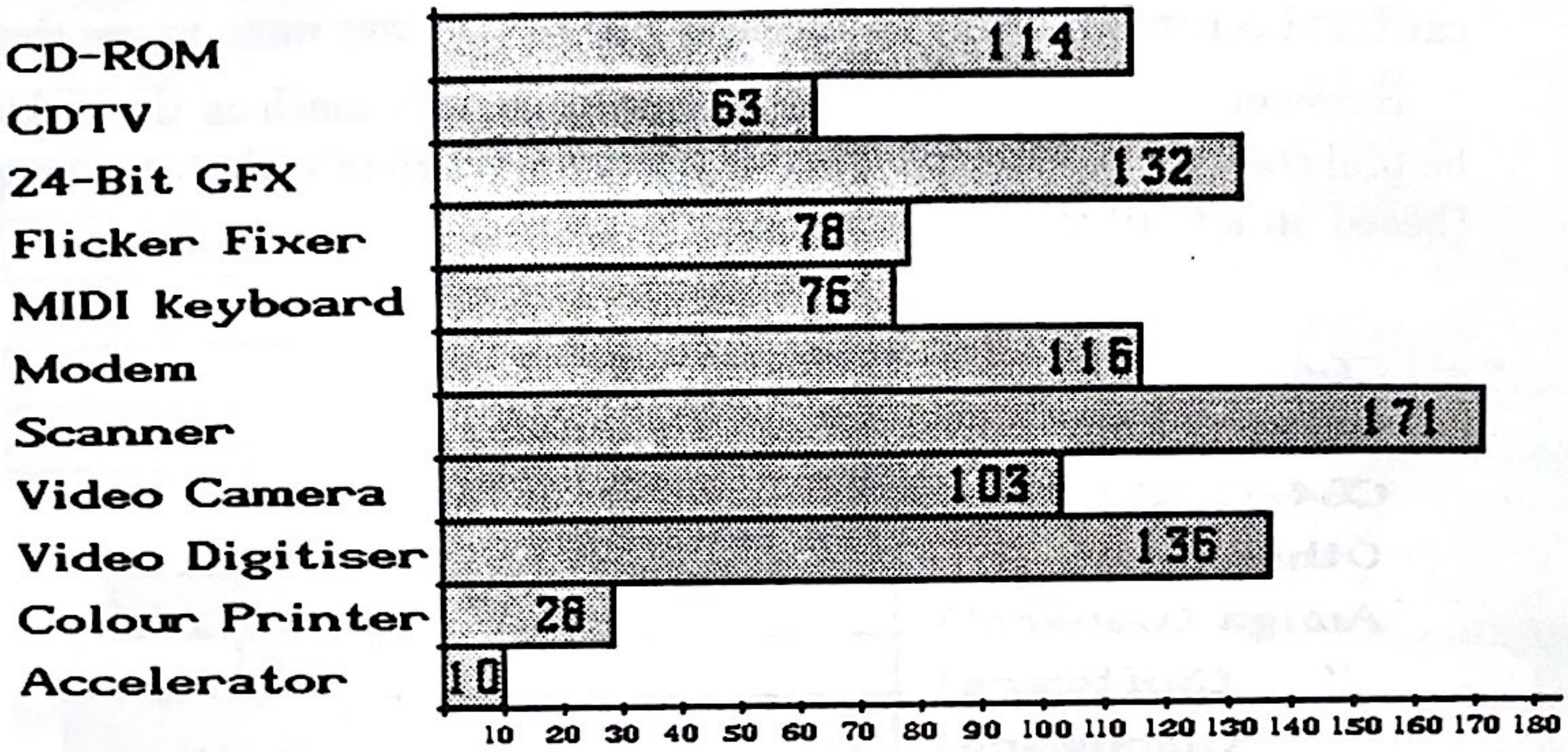


*Figure 8: Allocation of Funds—Hardware*

who should access it—half want it public and the other half are divided over CUGI members only and Commodore Users. Seems to me that we make it public, keep only Commodore News on it and have extra facilities for our members. On the other hand, 73% of the membership have never registered with Shareware authors—embarrassed silence or what?

The Newsletter is definitely a hit! 17 out of 22 claim *It's a must!* and 4 others reckon *It's a good thing.* I can only assume that the other one couldn't read! 55% of you read it all and 75% claim to read all or most. No-one said it was *Over their head* although one bright spark reckoned it was over Brian's head! While some credit must go to the Editor, its success is due to its contributors and over half of you can smile contentedly. A further quarter intend to contribute in the future thus guaranteeing the Newsletter's continued success. Well done. All that remains is to add more pictures and, to a lesser degree, more games reviews, more hardware items and even colour. Any ideas on the matter of pictures would be appreciated—over half of you want more pictures.

Finally, the discount scheme is known by most but not all members and over half have already used it with more to follow soon. If you know of any item that we might be able to get discounted, mention it to the Discounts Officer.

So there you have it—CUGI's vital statistics!! Now it's back to the committee to see how the group can best meet your requests and needs. Thank you very much for your time and ideas. We'll be in touch. Cheers.

# Denver Devcon Notes

Last September, Commodore hosted the annual North American Amiga Developers Conference in Denver. Devcon provides an opportunity for developers to meet with Commodore and each other, preview new Amiga technology, attend programming seminars, and generally have a ball for a few days. The following official text file was released to the press shortly after the conference finished.

## Devcon Statistics

- 265 attendees from six countries:
  U.S.A
  Canada
  Germany
  France
  Austria
  Australia
- Over 800 pages of documentation
- 4 disks of example code
- 1991 is the first year that Commodore has hosted back-to-back developer conferences in the U.S. and Europe. The objective is to provide the most current information to the entire developer community.

## USA Marketing

David Archambault, Director of Marketing for Commodore Business Machines, U.S. reviewed current and future marketing plans for the Amiga. David announced that the A3000 Power-Up Program has been extended until October 31, 1991. The plan offers a trade-up value towards the purchase of an A3000 of up to $1500 on any 8-bit computer system sold by Commodore. Also announced were print advertisements scheduled for fall placement that target to the professional video and presentation markets.

Mr. Archambault also brought the developers up to date on the current status of the Amiga market:

A3000s are substantially outselling the A2000. The major thrust of Commodore's professional marketing will be on the A3000.

Two vertical markets, professional video and presentation creation are the focus for current advertising and promotion campaigns. Commodore has invested more market resources to the video segment than any other vertical market, an effort that has had dramatic success—Commodore can claim up to a 70% market share of the professional video market. Commodore's market research shows that 15% of A2000/A3000 buyers purchase their unit for presentation creation. This shift in product use has been seen over the past twelve months.

Commodore Express (the joint customer service organization of Federal Express and Commodore) covers A500 sales, as well as the high-end A2000 and A3000. The customer service provided by Commodore Express has reached a 99% satisfaction rate with callers.

New marketing programs under development for the A500 are aimed at current 8-bit users, educators and parents of children with game machines, and center on a lower price point. One of the benefits expected is a future increase in high-end unit sales, as many A500 owners eventually "trade up" to the A2000/A3000 series.

Programs for the education market will include continued contact with educators and students already made aware of the Amiga. UNIX will continue as a key thrust of educational marketing, as will courseware and training based on CDTV and AmigaVision.

Commodore U.S. will be aggressively targeting the training and kiosk markets, as a substantial opportunity is seen for multiple unit sales. Currently in development is an advertising and promotional campaign targeted to Fortune 1000 human resource and manufacturing professionals, typically the two top users of computer-based skills training materials.

## CDTV

Members of Commodore's Special Projects Group spoke at eight devcon sessions. Gail Wellington, Director of Special Projects, reviewed current marketing plans with the developers. Promotion in the U.S. will include a major launch in key test markets, the largest being San Francisco. An aggressive television and print advertising run is scheduled along with a direct mail campaign. A continuing geographically targeted marketing campaign is being planned. It is expected that 10-15 major markets will be launched in the U.S. by Christmas.

A similar campaign, on a national level, is being planned for the United Kingdom. During a recent U.K. promotional campaign, fully 80% of respondents who visited the stores carrying CDTV purchased a unit.

Ms. Wellington also announced Commodore's plans to bundle Groliers New Electronic Encyclopedia and Psygnosis' Lemmings with every CDTV unit in the U.S. Groliers New Electronic Encyclopedia is one of the best selling CD-based titles, according to industry analysts.

### Release 2.0 Enchancer Kits

One of the most popular announcements at the Amiga Developers Conference was Commodore's plans to make AmigaDOS Release 2.0 Encancer Kits available starting in October. In the United States, a full update to Release 2 (including Release 2 Kickstart ROMs) will be made available through Service Centers. Manufacturing plans include a controlled production phase-in of Release 2 on all current Amiga models.

Commodore has worked closely with Amiga software developers to ensure compatibility of software with Release 2. Over 300 Amiga software titles have signed up for Commodore's "Release 2 Compatible" sticker program (for a list of titles, contact: Beth Rodriguez, Commodore Applications and Technical Support, (215) 431-9100).

---

## Devcon Keynote Address

*(This is a transcript of the opening address given to developers at Denver by Jeff Scherb, Vice President of Applications & Technical Support at Commodore U.S.)*

A lot has happened since we were together last year at the Atlanta Devcon. I spent a great deal of time thinking about this year's keynote address, and in going over the events of the past year, I was very surprised to see how much we at Commodore and you as developers have accomplished. We often lose sight of the progress of the war because of our particular daily battle, and I think it's worthwhile to take a few minutes to review what we've accomplished over the last year.

It's been a very good year for us:

- We are now getting the press coverage we deserve. The Amiga is finally being recognized as a powerful and cost-effective alternative to the PC and Macintosh. We are getting noticed by the professional computing community.
- The A3000 is shipping in volume. The recent "A3000 Power Up" sales promotion run by Commodore in the US has exceeded all of our expectations, and has, in fact, resulted in a backlog of orders for the machine.
- The A500 continues to gain momentum, particularly in Europe. Many believe that the 500 is now poised to repeat the success of the Commodore 64.

- CDTV is shipping, to tremendous press reviews. Over 100 titles are already available for the machine. The press now defines the "interactive multimedia player" machine generically as CDTV, rather than CD-I as it did a year ago. We are now the leader that the "other guys" have to follow.
- We are now shipping the Amiga running AT&T System V, Release 4 UNIX. This is one of the earliest commercial implementations of V.4, and our implementation includes Open Look and color X Windows running on the University of Lowell graphics card. Over 150 applications are already available for the machine.
- The A3000T "tower" machine is now shipping—this elevates Amiga expandability and performance to new levels.
- AmigaDOS 2.0 is now finished, and ROMS are being manufactured.
- Over 1000 developers are registered in the CATS-US Developer support program. Over 500 are registered in the support program in Europe. This is a significant increase over last year.
- There are more developers attending this year's developer conference than last year.
- Commodore continues to be profitable, and just closed the fiscal year at over $1 billion, for the first time since the '64 days.
- In the next month or two, we expect to ship the 3 millionth Amiga!

So we all have good reason to reflect back on the year since the last developer's conference and feel very good about our accomplishments.

There's much more coming, in terms of new hardware and software, and we'll tell you about some of them over the course of this week's conference. I won't steal the thunder of the session speakers by giving you any details now. Let's turn to the present—where should you developers focus your efforts to make 1992 an even better year?

- First, make sure you are strong in the areas of the world where Commodore is strong. You will see from our annual report that Commodore does about 85 percent of its business in Europe. If you don't have good European distribution, or your product is not available in the local European languages, you are missing out on a very large market.
- Support AmigaDOS 2.0 now! The time has come. We will be shipping machines with 2.0 in ROM very soon, and the time for you to release 2.0 compatible upgrades is now, if you haven't done it already.

- Continue your focus on quality. In general, Amiga software is of much better quality now than it was a year ago, but we all need to continually focus on quality.

CATS is here to help you. That's the only reason we exist. Please take advantage of the services we offer. This year, CATS has released more developer tools and documentation than in any other year. Here are a few examples:

- The User Interface Style Guide will help you create more intuitive, easy to use, and standard user interfaces, for your applications.
- The Application Installer, available from CATS for free distribution with your applications, provides a standard and simple way for your users to install their applications on their hard disks. Use it. Plan to attend the session this week covering this new tool.
- AppShell, a development tool designed to serve as a sample application and a basis for the event processing code in your applications, is available now. Take advantage of this tools to speed you application development.
- The ARexx development guide will help you add this important interprocess language capability to your applications. You can buy this manual at the Devcon Store this week.
- We now have *AmigaGuide*, a hypertext development and runtime system, available to you for inclusion in your applications. Using this tool, you can easily add hypertext-linked help text to your applications. Be sure to attend the session on *AmigaGuide* to learn all about this new tool.
- Most of the 2.0 ROM Kernel Reference Manuals are finished and either are available now at the Devcon Store, or are in the final printing stages at Addison-Wesley.
- How many of you are using *Enforcer* and *Mungwall*? Why didn't everyone raise their hands? Every developer should be using these debugging tools as part of their normal routine. MS-DOS and Mac developers can only dream of powerful tools like this, since they can't be implemented on those primitive computers. Make sure you take advantage of these tools.

CATS has quite a few other projects in the work to help make you more productive in developing applications:

- As I said before, if you're not in Europe, you need to be. *Crossing Borders*, a new reference book to be available from CATS early this fall, takes a "cookbook" approach to getting to Europe. How do you get distribution? Support? Translation of your manuals and programs? *Crossing Borders* is the manual that will help you answer these questions. Make sure you get your copy as soon as they're ready.

- We're putting the Amiga developer docs on CD-ROM, using our own *AmigaGuide* hypertext system. Using this you'll be able to integrate this online reference to the Amiga operating system and hardware into your favorite text editor; you can call up OS function definitions while you're programming, and using the cut-and-paste features, move examples directly from the documentation into your code. This should be available sometime this fall, and you can see a demo of it this week in the hardware lab.
- We're arranging special Developer/Distributor meetings to help you meet the European distributors and gain distribution in Europe. The first of these will be held this year at Amiga '91 Cologne, the largest Amiga show in the world. Last year, this show drew 60,000 people. You'll be getting a mailing from CATS with information on these meetings next week.

We've got a great Devcon planned for you; let's look at some of the highlights:

- Several sessions on new hardware are being held today;
- We have a lot of detailed CDTV sessions for you;
- A presentation by a translation service that can help you move your applications to Europe;
- Several sessions by Commodore's US marketing department, to help you understand where we're going from a marketing point of view in the US;
- A session on finding the right European distributor, by CATS Applications Manager, Europe, Wolfgang Trompetter;
- And many other technical hardware and software sessions.

Enjoy the conference; make the most of it; CATS is here to help you.

Thank you.

# Timebase Correctors Revealed

*by Colm O'Rourke*

With all the news coverage surrounding the Video Toaster and similar devices, most Amiga users have probably seen the phrase *Timebase Corrector* mentioned. But what exactly does this mysterious piece of equipment do, and why might you want one? To explain this, we need to go back to television basics.

A TV picture is made by a beam from an electron gun which paints lines across the phosphor screen of a TV tube. After each line is painted, the horizontal sync pulse switches off the beam while the gun returns to the left edge of the screen to begin painting the next line. When the gun reaches the bottom of the screen, the vertical sync pulse switches off the beam and returns it to the top of the screen where the process begins again. This time the beam fills the lines between the ones it painted on the first pass.

The first set of lines is called the odd field, the second set is called the even field. One odd plus one even field makes a frame. A PAL frame (625 lines) lasts 1/25th of a second. The heads on a video recorder drum move diagonally across the moving tape laying down magnetic trails that represent the video signal. One sweep of the head contains one field of video information. When the second head comes around the tape has moved slightly bringing the second field into line.

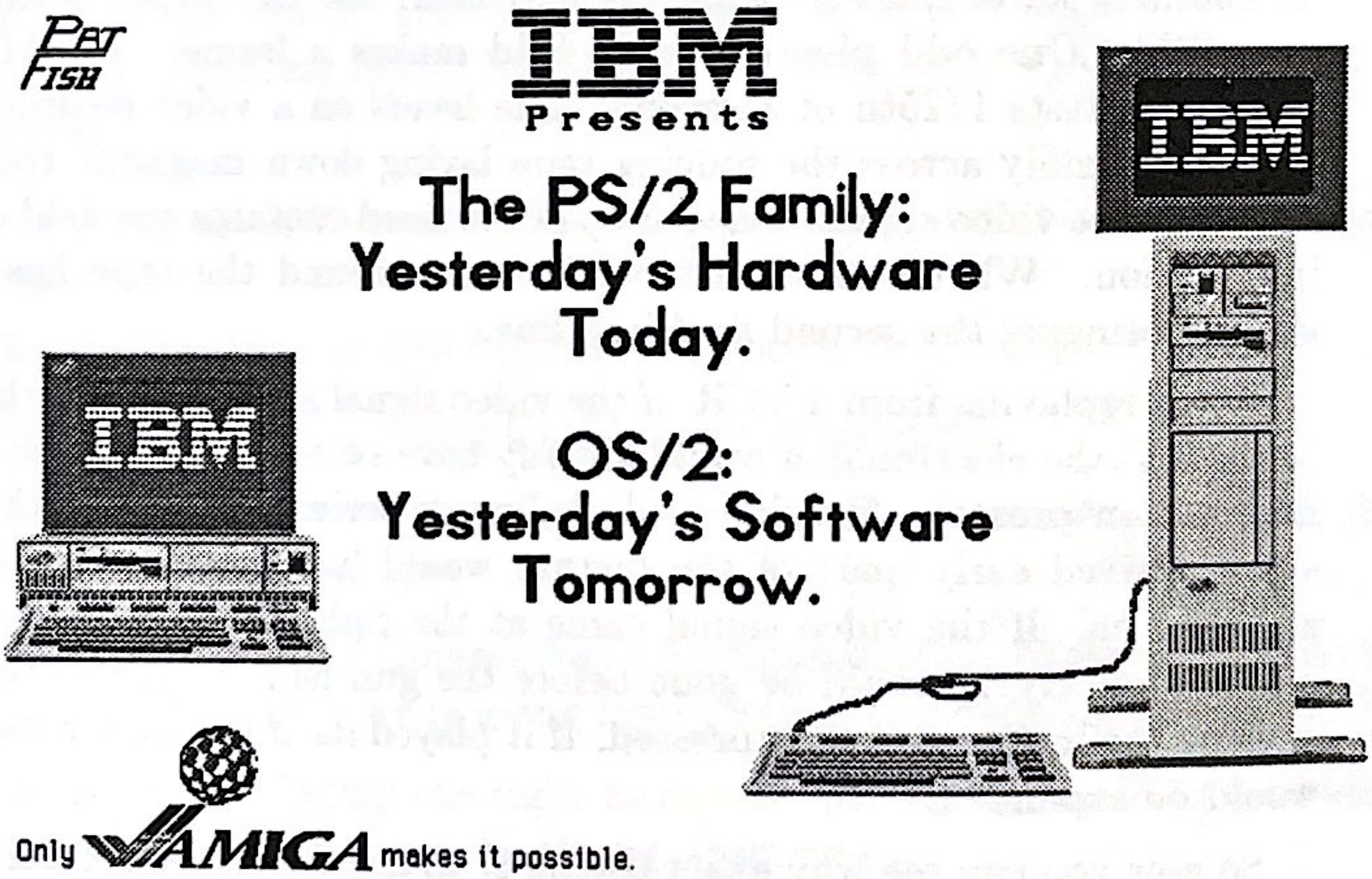
When replaying from a VCR, if the video signal arrived late at the input to the TV, the electron gun would already have switched on with no picture information present. Result—a black line or series of lines. If the video signal arrived early, part of the picture would have gone before the gun switched on. If the video signal came at the right time but played itself out too quickly, it would be gone before the gun had completed its sweep causing the image to be compressed. If it played itself too slowly the image would be expanded.

So now you can see why exact timing is all important. A normal TV set can handle small timing errors but an editing recorder will most definitely not. Why are video recorders so inaccurate? Tape stretch, variations in temperature and humidity, and mechanical wow and flutter can all cause the tape speed to vary.

A timebase corrector acts as a buffer which outputs a smooth flow of video no matter what timing errors occur at the input. If it can hold 625 lines of video, it is called a frame store and can in theory correct any amount of error. A framestore TBC is said to have an infinite window and will store the current frame until the next useable picture is presented to it. Perhaps you have seen this on television where a mobile camera link is being used on the back of a motorbike or on some other unreliable signal platform where the sound continues as normal but the picture pauses briefly then picks up again. This is a very good example of a TBC in action.

The TBC is a useful tool which will help you to produce a consistantly reliable standard of work. Will a TBC convert from NTSC to PAL? Unfortunately no. To achieve that piece of electronic magic, you need a standards converter which is very expensive indeed.

---



# The A2320 Display Enhancer

*by Karl Jeacle*

When I first got my Amiga, I was pretty happy. I had a nice setup. A2000, bridgeboard, partitioned HD... ah yes, this was the machine for me—what on earth had I been doing with a PC for that long?

But as time passed, I started to get frustrated at the limitations of my setup. There was more I could do with this machine—if only I had a larger hard disk, some more memory, a faster modem... and wow, wouldn't a flicker fixer be nice...

Unfortunately, flicker fixers were about Stg£325, and a multisync at least that much again. A bit out of my price range. However, at the start of this summer, Microway reduced the price of their flicker fixer to Stg£125! They claimed this was due to the boards selling so well; however, a more likely explanation is that is was due to competition from Commodore's A2320.

## The A2320 Display Enhancer

The A2320 is essentially the same video circuitry that's inside the Amiga 3000 which eliminates the irritating 'flicker' you get when in interlaced modes. It comes on a board which fits into the video slot of an A1500/2000.

The board arrives in a compact (26 × 17 × 5 cm) box (small enough to dodge customs) which includes a *Test/Demo* disk, 26 page manual, and fine-tune adjustment tool. The demo disk contains an impressive collection of both interlaced and non-interlaced pictures to test your board.

## Setting it up

Before installing the A2320, you must check whether your Amiga has a 2 or 4-layer mother board. This is determined by looking at the front left-hand corner of the motherboard. You should find a label that says "AMIGA B2000—CR" which indicates that your motherboard does indeed have a 2-layer motherboard. If you are unfortunate enough to have an old Amiga with a 4-layer board, you are instructed to get a replacement motherboard from your local authorised Commodore service center.

However, provided you've got the right motherboard and you've got a suitable monitor, installation should be a breeze. Just plug it into the video slot, and away you go. On the back is your DB15 VGA style monitor connector, an enable/disable switch and a fine tuning adjustment screw. The enable/disable switch controls the frequency output from the DB15 connector. This does not affect output from the existing DB23 video port on the Amiga.

The first time you use your new toy, you'll probably notice some slight jittering on the display; a quick twiddle of the fine-tune adjustment tool soon puts this to rest. A couple of "testcard" pictures on the demo disk are helpful here.

## Inside the A2320

When enabled, two things that the A2320 does are:

- De-interlacing: While in interlaced mode, this removes flicker.

  The reason an Amiga flickers on a high res screen is because it first outputs all the even display lines followed by all the odd display lines. It does this so fast, we can see a full image on screen, but it appears to flicker. The display enhancer output frequency is twice that of a stock Amiga, so with the help of 384Kb of on-board memory, it can output both the odd *and* even display lines at the same time, thus eliminating flicker.

- Scan-doubling: This removes visible scan lines in non-interlace mode.

  This is achieved by drawing/refreshing the screen twice as many times as usual giving thus giving a rock steady display without any scan lines. While de-interlacing, the display enhancer uses its high frequency output to display both odd and even display lines; here, it just outputs the same display lines twice. This is known as *double-scanning.*

The display enhancer can work in both NTSC and PAL modes—handy if you have an ECS Agnus, and want to work in NTSC. Since we're in Europe, the board will be delivering PAL frequencies: Vertical PAL frequency is fixed at 50 Hz. Horizontal PAL frequencies are 15.625 kHz (A2320 disabled), and 31.25 kHz (A2320 enabled).

15.625 kHz is the frequency that a stock Amiga outputs, driving your standard 1084/8833 monitor. It flickers in interlaced mode. The A2320 outputs this frequency when you flick the switch to disable mode.

31.25 kHz is the frequency that the display enhancer outputs when enabled. It does *not* flicker in interlaced mode, but is too high a frequency for a 1084, requiring a multisync quality monitor.

## Choosing the right monitor

There are basically three types of monitors you'll come across—standard 1084/8833, SuperVGA/low-end multisyncs, and high-end multisyncs. Here are sample specifications for these types of monitors, along with the A2320 output:

| *Monitor Name* | *Description* | *Vert. Freq.* | *Horiz. Freq.* |
|---|---|---|---|
| CBM 1950/NEC 3D | High-end Msync | 45→120 Hz | 15→36.5 kHz |
| NEC 2A/My SVGA | Low-end/SVGA | 50 Hz | 30→40 kHz |
| 1084/8833 | Standard monitor | 50 Hz | 15.625 kHz |
| A2320 | Enabled | 50 Hz | 31.25 kHz |
| A2320 | Disabled | 50 Hz | 15.625 kHz |

As you can see, a 1084 simply won't cut it. It can only handle a disabled A2320 output—pretty useless.

A high-end multisync on the other hand has no problems. It will happily take a flicker free 31.25 kHz signal from the display enhancer. It will even "scan down" to the original 15.625 kHz Amiga output if you disable the A2320. This is useful if you're considering buying something like HAM-E. Go for a monitor like this if you can afford it.

If, however, you're a little short of cash, you can get away with a low-end multisync which will only recognise an enabled A2320. For most applications (except HAM-E and other such devices) this is very acceptable. I managed to get my hands on a SuperVGA monitor which has a variable horizontal frequency of 30→40 kHz which works just fine.

However, I must stress that many VGA/SVGA may NOT work fully* with the A2320. Many VGA/SVGA monitors can only handle a 31.5 kHz input, and this is not what the display enhancer outputs. So, if you are thinking of an SVGA, make sure you try it out with your system before you buy.

## Conclusions

The only real criticism of the display enhancer is the top half-line of flicker which remains on screen, even when the board is enabled. This occurs on the A3000 too. Many people have given out about it in the past, but they're really making a big deal over nothing. Just adjust the vertical stretch knob on your monitor, and it'll disappear off the top of the screen.

Once you get a taste of flicker-free Amiga, you won't ever want to go back. It gives the Amiga the professional feel it's been lacking—and if you've ever used Workbench 2.0 on a 3000, you'll know just what I mean: an interlaced Workbench screen with custom fonts and colors looks fabulous. The display enhancer is a great product, and I recommend it highly.

---

*** I say fully, since they may work in NTSC mode, but not PAL.**

# The Hacker

## *by Brian Ward*

Joe hated the word, but that's exactly what he was. He knew what he was doing was wrong, but what they were doing was even more wrong.

He remembered how it had all started—the ad in the newspaper, looking for a programmer for an astronomical fee that was too hard to pass up. The interview was very strange indeed. It was held in a hotel room, and he never saw the faces of the people who were interviewing him; he didn't care so long as he got the job, which he did.

Day one was also strange. Joe was told to report to an office on Monday morning. However, when he got there, instead of a plush, multi-storied complex, he found a bare office with a desk, a computer, two phone lines and two modems. He was met by a man who gave him his instructions as to what to do.

Basically, it was an E.F.T. (Electronic Funds Transfer) program. The banks use them all the time, and now certain banks were allowing their customers to do the same. All you had to do, was phone up the bank, day or night, tap in a few passwords on your tone keys, and you were in.

However, they (whoever THEY were) wanted a program to totally automate this whole proceedure. For example, they wanted the computer to dial up a bank in say Kansas (why Kansas? You couldn't get a more remote state if you tried) and transfer funds to a bank in Idaho (another weird choice of state). Again, had Joe thought about it at the time, he might have figured it out before now, but the money was great, and he had to pay for that new high-speed modem he'd just bought with his credit card.

So, Joe set about his business. He was to alter an existing program that his instructor said had been written by someone (he didn't say who or why they couldn't get his predecessor to do the alterations). Breaking into the code wasn't too difficult as the software wasn't written for commercial release.

One of the first things he noticed was the date the code was written. It was less than a week old. No, it couldn't be—and yet there it was... (c) Bob Newman 13 September 1991. The name rang a bell, but he couldn't place it. It was very recently that that name had cropped up somewhere, but he was damned if he could remember. Once into the program, he dumped the code out so he could study it at home.

Joe was given a set of phone numbers to get the computer to dial up and transfer the funds. After dialling the first number, he tapped in the pass-word and hey presto, he was into the E.F.T. area of the bank's computer. He inquired as to the balance and got the second bigggest shock of his life: the balance read $10.9 million.

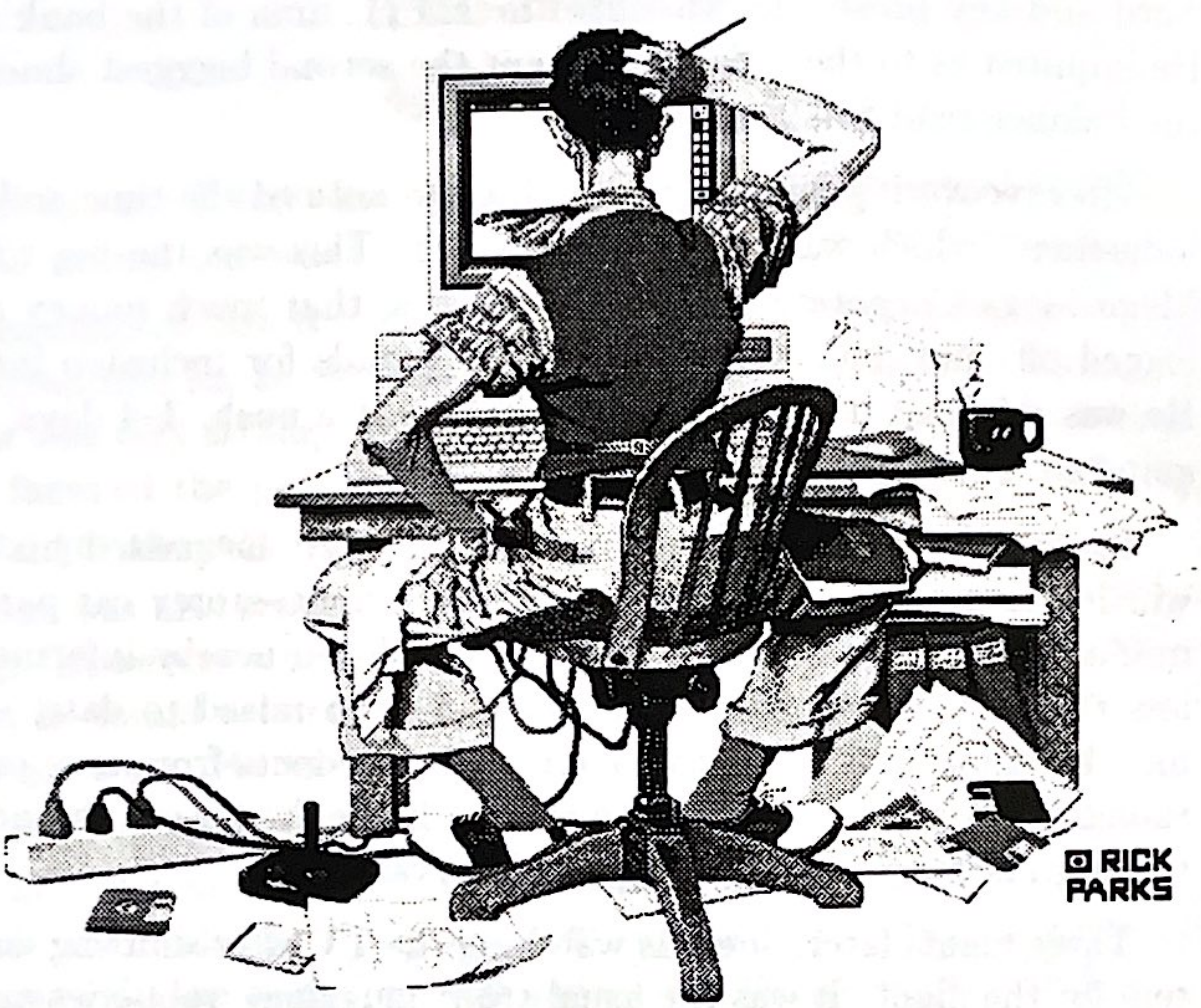
After recovering his composure, he also noticed the time and date of the lodgement, which was 20 minutes earlier. This was the big time. Only a Mega Mega Corporation would be moving that much money around. He logged off, and made a note of all the details for inclusion into his code. He was thinking that the job would take, at a push, 3–4 days, but he was going to try and milk it for all it was worth.

Joe left for home around six. On the way, he passed his old school, which was in danger of being made into a multi-storey car park unless $2 million could be found. A large chart displayed nearby informed him that less than a quarter of that amount had been raised to date, so it looked like the school was going to have cars for students from now on, which he thought was very sad. When he arrived home, he logged on and continued to program away, saving at regular intervals.

Three nights later, Joe was watching the TV after shutting down his system for the night. It was the usual stuff: muggings, robberies and murders. However, one particular unsolved murder gave him the biggest shock of his life.

The reporter was standing beside a river with police all around him and he was saying, "... today. He was found by a jogger early this morning. He has been identified as Robert Newman, a computer programmer, who was formerly employed by Scott Corp, which is rumoured to be a front for drugs baron Slim Jim Matassa. Newman's wife reported him missing three days ago when he failed to return home after leaving for a meeting with three men earlier the same morning. Slim Jim is currently under investigation by the IRS and the DEA but as yet they have no concrete evidence for an inditement. Apparently, any money purportedly received from drugs just disappears, a DEA spokesman told me today."

It all fell into place at once, and with it came tunnel vision. It was as if all the world was now watching him waiting to see what was going to happen next. "I'm dead before I've even had a chance to live," Joe thought. However, it suddenly occurred to him that only he knew all the facts at this moment in time. He was currently writing a program for Don Matassa, to move drug money around the world very very fast, and the way he was doing it left little evidence behind. And another thing... Newman must have been creaming a little off the top every so often, and had been caught. But where was the money? Only Newman knew where he'd sent



it, and now that he was dead, no one would ever know. There and then, Joe decided that he was probably going to be killed when the job was done, so he was going to bring Matassa down with him. But how?

The following day was Saturday, so Joe didn't have to work. He logged in at about 11:30 a.m. and was met by a very peculiar greeting. It read as follows: "Hi, I'm Bob Newman and I'm probably dead... see if you can resurrect me. You've got 24 hours."

Joe knew that Newman had left a time bomb in the computer that was set to display this message if he didn't log back in within a specified time, but why leave a message like that? Unless he wanted someone to know what he had been doing. *So, what would I do?* thought Joe. *Let's see... I'd leave the time bomb like Newman did, but how would I get the information to the person? I'd leave it on the hard disk!*

Joe did a directory of the disk; nothing obvious showed up. *No, he wouldn't leave it somewhere anyone could find it; he also knew that they would have to get someone to finish the program, so it would have to be someone who knew about computers. Of course! He'd hide the file!* Joe ran a utilities program to look for hidden files. None found.

He left the screen, and made himself some lunch, during which he thought of nothing else. "What the hell would you have done with the information!"

he exclaimed. It was then he remembered something else: the last piece of the message read that he had 24 hours. 24 hours before what... before a virus would entirely corrupt the data on the drive making it unreadable for ever.

*By God you were cute*, thought Joe, realising that he was running out of time. He knew that Newman was very clever and that he'd planned this for quite a while, so what he had amassed must also be substantial and no one knew how much or where. It was the perfect crime, except that he'd been caught. On the other hand, Joe was still in the clear. The money was his when he found it. IF he found it.

Joe re-read the message. "Hi, I'm Bob Newman and I'm probably dead... see if you can resurrect me." Resurrect me? He was dead, how could he be resurrected? He was gone, erased, deleted from this world, so how could... suddenly it dawned on him.

He called up the menu on his utilities package again. This time he told the program to search for deleted files. It came up with several, but one stood out. He had simply called it *Lazarus*, as in risen from the grave, or resurrection. Joe restored the file, which turned out to contain text. He downloaded it to his own machine, and then used the utilities package to kill it off the other machine, making it permanently unrecoverable. Then he loaded the file into his wordprocessor. It was the file he had been looking for.

He started reading. "Hi once again. As I've already said, I'm Bob Newman, and I'm probably dead now. The reason you see is this. I was working for Don Matassa for several years, but I'd decided to turn States evidence in return for a new identity (ha ha, New-man, geddit? Sorry, it's my last joke, you know). Anyway, I thought I would take a little cash with me 'cos these witness protection programs don't have a very high budget. The fact you're reading this means I've either been run over by a bus, or am currently feeding a lot of fish at the bottom of the bay.

"Anyway, it's immaterial now 'cos it's not me who's currently at the screen, and I have set in motion a virus that will encrypt the data within 24 hours of you first reading the message that lead you to this file. That doesn't leave a lot of time to do the things I want you to do. Why should you do this? Well, would $25.8 million help? Yup, that's what I had at the time I wrote this, it could be more by now. If you are interested, continue reading. If not, for your own sake, don't read any more and just let things take their course, but get outta town for a long while, 'cos the stuff's gonna hit the fan soon. Okay, you're still with me. Then we'll start."

Newman then went on to explain exactly where the money ended up and how it got there. It went through several states and then ended up back here in the city. He gave all the bank account numbers, sort codes and most importantly, the passwords. He then outlined his plan. It was going to take all night at best, and he'd need at least 24 hours to disappear.

That was how Joe had ended up hacking into about 20 different computers on the same night. The fact that he didn't actually have to crack a system due to the fact tht he had the passwords didn't make him feel any better. First, he logged back into the office computer. Then he instructed it to dial up the various different banks.

The way the system was supposed to work was like this. In each city, there would be a computer in a room on its own, just like the one he was connected to. At a given time, another computer would automatically ring it up and pass on the amount of the transfer, what bank it was currently in and what bank to send it to. The whole country was criss-crossed with a network of these computers, all of which would have Joe's program installed in them, moving money at the speed of light all over the place. By the time it returned to the city, the money would have passed through several banks, making it a little cleaner each time. The banks were running a laundering service and they didn't even know it.

The 20 different computers were all in the the city, and they were the banks where the cash ended up. Strangly enough, they were all branches of the same bank. Newman had said that the bank was controlled by Matassa, but no one would ever have been able to prove it.

The 'Ace in the Hole' that Newman had was Matassa's personal bank account. In the notes, Newman had said that this was the hardest part of his plan. He had searched all over the city for weeks looking for the account, but could not find it anywhere. Then, out of the blue Matassa had given him a personal cheque for money that he owed Newman. He couldn't believe his luck.

Now Joe was transferring millions into an account that held about $200 just minutes before. The 24 hours was drawing to a close as Joe transferred the last of the accounts into the Don's account. The last thing he did was to upload a file to the confidential police telephone line. It wouldn't take them long to realise that the high pitched whine on the recording, was in actual fact computer data which they would have no trouble in unscrambling.

In case of line noise, he uploaded it again, just to be safe. The message just gave dates, times of drug deals and the banks that were being used. All this information had been gathered for years by Newman, just in case the Don decided he was expendable. The police would freeze several bank accounts on Monday morning pending investigation.

A week later, Joe was in a hotel room looking at the TV and the same man that had announced Bob Newman's death was announcing Don Slim Jim Matassa's arrest. Matassa looked bewildered, as if he had lost something. Somehow, over $200 million had mysteriously appeared in the Don's personal account overnight. The problem was that no one was missing that kind of money, so it could not be a computer error. He was not a happy puppy.

Joe was very happy. He had $29.2 million dollars scattered all over the country in bundles of $50,000, deposited in accounts already set up by Newman. One of the first things he had done with his new found wealth was to transfer $3 million into the fund to save his old school. (He thought that they might need some loose change to buy some new equipment, so he included an extra million to be safe.) He felt pretty good about this, and would have loved to see the developer's face when he found that the school had been saved. *Chalk one up for the little people*, he thought.

Joe felt so good about the school that he decided to play Santa Claus. So it was that over the next few months, several needy charities suddenly received money from an unknown source to bail them out just in time. They were all convinced that it was the work of God, but Joe knew better. It was a bit like Robin Hood or 'robbing hoods,' he mused. He decided that Newman would have appreciated that joke.

Whenever he needed money, all he had to do was log in to one of his acounts and transfer the funds to the account of the charity. He was tempted to take some for himself, but it would have been very hard to explain to the bank manager, so he decided to leave well enough alone. What he needed to do now was to find employment. Joe hoped the new job would be different. He'd had enough excitement to last him for the rest of his life.

# Graphics Events, Part 2

### *by John Pritchard*

Where were we? Oh yes, awaiting a return call from Commodore regarding my crashed-out, de-initialised, infuriatingly tempermental Amiga 2000. The most expensive piece of self-torturing, soul-destroying, mind-melting gear I've ever bought. I've now no friends, no social life, the family wants me locked up and I've aged... oh how I've aged!

Can you explain to me how a seemingly normal adult, whose favourite pastime is to be slowly basted by holiday sun and saltwater on a Spanish beach, could spend the price of six such holidays on a device which instead of rejuvenating him, makes him age visibly, turn a ghastly shade of puce and develop a requirement for new, more powerful eyeglasses every three weeks? Nor can I!

Anyroadup and to my astonishment, someone from Commodore actually returned my call. Not only that, he sounded intelligent, informed, and to cap it all, interested. I just knew that such behaviour marked him down as being foreign! Sure enough, his name was Sajjad!

Well, I explained all and he, being as puzzled as the rest of us, promised to give the Amiga the third degree if I would return it pronto. This I arranged with our local courier service, paid an extra fiver to ensure delivery before 9 am, rang Commodore to confirm this and they promised to be waiting anxiously for its arrival with resuscitation equipment, electron drip and pixel pacifiers at the ready!

I rang at four that afternoon to enquire about the patient's condition, only to find no one knew of the planned (and confirmed) arrival, or of its whereabouts and that Mr. Sajjad wasn't there. We were safely back on familiar ground—even Mr Sajjad's previously aberrant practices had now reverted to standardised British. So much for me ensuring early delivery! My extra fiver had merely ensured that the Amiga sat on a Commodore bench all day, unloved, uncared for, unwanted. I could have kept it here for that day and sat glaring malevolently at the stupid thing.

Eventually I was assured that having found my machine they had re-created my animation in the most demanding form possible, that is to say as a ray-traced anim, and had experienced no problem. When I got it back the disk drive unit had been slightly bashed in and its front panel kept falling off but I had no stomach left for further disputation. When I could find no evidence of the successful test anim and rang them about it, they said they had accidently wiped it off.

During the course of these proceedings I was not alone in being suspected of anti-machine subversion. So was the electricity supply. First it was 'spikes' (sudden bursts of excess power). So we installed an anti-spike doofer (do fer now) into the mains feed. The next suspect was 'Drops' of power which means drops of power. I took the Amiga home and used it only at dead of night when no one outside of Frankenstein Towers would be putting loads on the mains.

We now had reached the classic situation:- the manufacturer had no problems; the dealer had no problems; I was obviously crackers. Try to prove you ain't. S'impossible.

Gingerly I started back to work with the beast slowly increasing the complexity of its tasks. I am tempting fate by saying that I've now had three months trouble free computing though I'm still keeping crossed fingers, eyes and legs, just in case. I cannot believe that I've not had a single problem in three months. In moments of post anguish paranoia I'm wondering if Commodore didn't just think "To hell with it" and simply installed an IBM inside an Amiga casing! I ask you, what can you rely on these days?

Why you may ask am I writing of these events? Three good reasons present themselves: to inform the unwary of these hazards; to console other sufferers by letting you know that you are never alone in your suffering; and to simply let of steam.

It's tempting to list the three megabytes of outrageous language I strung together during the worst moments or to reveal my assassination plans for manufacturers, dealers and particularly those all to familiar smug cretins who can't resist stating, "Oh I've never had those things happen to me." This merely spurs me on to plotting to fill their empty little lives with some new problems to be getting on with. Like delivering a hard-boiled egg filled with NitroGlycerine and a spoon to tap it with. Or a Cider and Cyanide "Pick me up" not to mention "Lay me down."

But my sweet charitable nature always wins over in the end and I quickly revert to positive thoughts such as what advice can I give to my fellow parishioners? The advice is based on certain observations. Here are a few of them: If you are an American and your new toy goes wrong, you trash it and buy another one... because you are an American... you're rich. If you're a West German and it goes wrong, you demand service and get it... because you're German and Germans get service (Ve haf vays...). If on the other hand you're a Brit and you get problems, then you pretend you don't—"What would the others say?" Stiff upper lip, salute the flag, Maggie might be listening! If the problems get worse then you prove your true Britishnesss by becoming a 'Whingeing Pom'. Nudge nudge, wink wink "Don't think this is terribly good with all this smoke pouring out of it, do you?"

If things get even worse, the next step is to grovel, cap-in-hand to the dealer who retorts, "Commodore made it, talk to them. Anyway no one else has ever complained. You MUST be doing something wrong. I'm a dealer and I've enough to do counting these cardboard boxes without having to learn what's inside the flamin' things!" As a last resort you nervously and apologetically ring Commodore... and I've already told you about that!

But whatever you do as a Brit, you won't ever rock the boat, shout, be rude, or even complain loud enough for the D.E.A.F. to hear (Dealers of England Anti-customer Federation). You are always, "The only one who has ever complained. The rest of the world is happy, so why don't you stop moaning? And if you think your measly three thousand pounds goes far in having an extension built on to my Porsche, then you're right behind the times Matey!" And we wonder why we get such poor service here.

## Advice

What advice can I give to you?

- Don't own any technical equipment.
- Don't get involved with video.
- Don't buy a computer.
- Don't get into computer graphics.
- Don't rely on others.
- Don't work for a living.

I think that's a pretty positive guide to people wishing to avoid potential bankruptcy of body, mind and spirit, not to mention bank balance. But if you like frustration, mindless machines that are out to get you, if you want irradiated skin, chronic eye irritation, a garret-like existence devoid of conversation, friendship or relaxation and a habit that makes heroin seem as addictive as Eccles Cakes, then buy an Amiga. You'll have to buy your own. Mine's not for sale. I'm giving it one more chance... one more... one more...

# Professional Draw V2.0

*by Colm O'Rourke*

If you want to produce drawings and text on paper to the maximum definition of your printer and you like the idea of being able to scale, mirror, rotate or distort your work while avoiding the dreaded bitmap jaggies, then *Professional Draw 2.0* from Gold Disk is the program for you.

PD2 uses structured graphics rather than bitmap graphics. Structured graphics organise an image into basic structures based on such geometric elements as lines, curves, ellipses and rectangles, producing a description of the image rather than the image itself. This description can then be magnified to any degree and the output device will draw the image to its maximum resolution without jaggies.

PD2 will work in interlace or non-interlace mode, and requires a minimum of one megabyte of memory. In addition to Amiga Preferences printers, HP Paint Jet and Xerox 4020, you can output drawings in PostScript format which allows you to use professional image setting machines with resolutions of up to 2400 dots per inch. PD2 also allows you to save your work in Encapsulated PostScript Format (EPSF), making it easy to export drawings to other packages.

PD2 facilities include lines, curves, squares, circles, magnify, move, measure distance, fill, blend, trace, layer, rotate, distort, grid, mirror, size, align text to curve, group, clone, line weight and line pattern. It is supplied with two fonts: *Triumverate* and *Times.*

PD2 is not a text layout program. It deals with a maximum of 256 characters at a time and each block of text can be treated in the same way as a structured graphic. Images can be positioned to an accuracy of one ten-thousandth of an inch. Gold Disk also include a remarkable program called *Trace* that can convert IFF bitmaps into structured drawings. This can provide a good start point if your artistic ability is lacking.

Last but not least, PD2 produces colour output on a suitable printer and also produces process and mechanical colour separations for professional colour work using selections from one billion definable process colours.

Pause for a moment and imagine the advantage of being able to go from concept to finished, colour separated, plate ready film for many design and publishing tasks. You can avoid expensive scanning, colour separation, stat camera and film stripping work. If you decide to go the 2400 dpi Linotron colour route, then consider getting a colour key or chromalin (a very accurate colour proof made from your colour separated film) before the print run starts because in colour printing, mistakes can be rather expensive if they are discovered after five thousand copies have been made!

PD2 is a powerful, easy to use program which does exactly what the marketing blurb says it will do. As a finishing touch, all images may be imported into *Professional Page*, Gold Disk's desktop publishing package, or into AmigaTEX (see below).

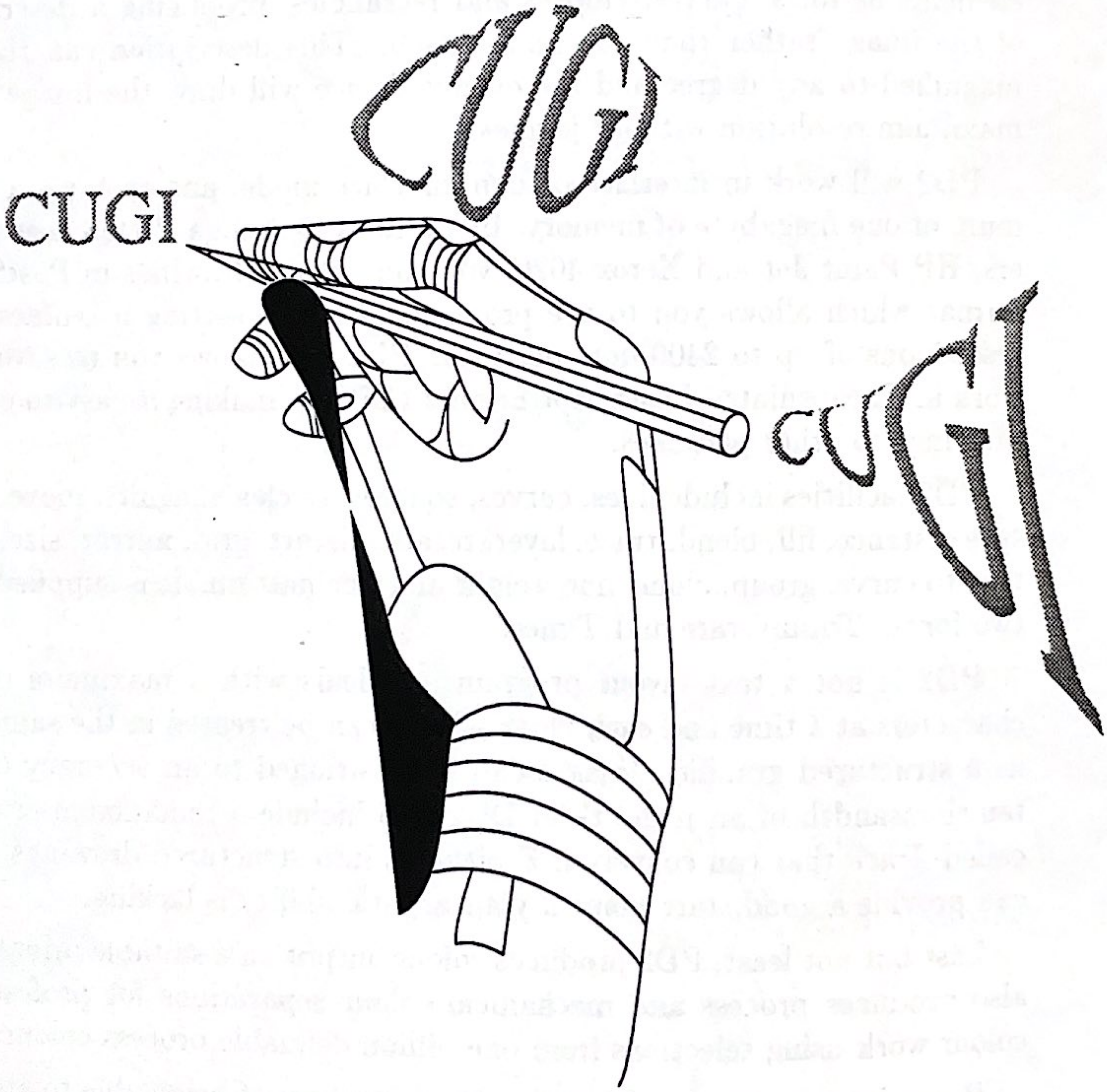


*Figure 1: Professional Draw makes special effects easy*

# Startup-Sequences

*by James Ruane*

Many of you reading this article already know what a Startup-Sequence is. This article is not aimed at you; it is aimed at those readers who have not yet begun to fiddle with the Startup-Sequence on their Workbench disks.

## What's a Startup-Sequence?

The Startup-Sequence is a script file (i.e. a file made up of CLI or Shell commands) that is executed when a bootable disk is inserted into the drive after turning on or resetting the Amiga. AmigaDOS looks in the directory called S on the boot disk, finds the Startup-Sequence file and executes the commands in it. The Startup-Sequence does many things on the average Workbench disk. For example, it loads the time from the battery backed-up clock (if one is present), sets the keymap on the keyboard according to the country you're in, loads the Workbench and installs a program to speed up text output.

## Okay, but what's all that got to do with me?

Well, with a little bit of effort you can modify the Startup-Sequence to run your favourite programs. The Startup-Sequence is stored as an ascii file and so can be changed by any text editor or word processor that saves as ascii.

As an experiment, find the Extras disk that came with your Amiga. Open the Tools drawer and click on the MEmacs icon. MEmacs is a powerful text editor, but it is not very user-friendly to novices. Therefore I would recommend a different text editor, such as Qed or AZ (on Fish Disk 346 in the CUGI library). Anyway, back to MEmacs. Select *Read-File* from the *Project* menu and type `S:Startup-Sequence` at the bottom of the screen. After a short while, you should see the text shown in figure 1 (minus the line numbers).

Let's go through each line one at a time. Line 1 adds 10 buffers to DF0:, speeding up disk access. Line 2 simply fixes some known bugs in Kickstart 1.2 & 1.3. Next, the current directory is changed to C: which makes it slightly faster to load in the remaining commands. After this, a message is output to the screen giving the current version of the disk.

Line 5 is for users who have expansion memory such as that in the A590, and also have an A501 expansion or similar plugged into their trapdoor slot. The expansion memory is faster than A501 memory, and *FastMemFirst* insures that it will be used in preference to the slower memory.

```
1  Addbuffers df0: 10
2  c:SetPatch >NIL:; patch system functions
3  cd c:
4  echo "A500/A2000 Workbench disk. Release 1.3 version 34.21*N"
5  Sys:System/FastMemFirst
6  BindDrivers
7  SetClock LOAD; load system time from real time clock
8                ; (A1000 owners should use DATE instead)
9  FF >NIL: -0 ;speed up Text
10 resident CLI L:Shell-Seg SYSTEM pure add; activate Shell
11 resident c:Execute pure
12 mount newcon:
13 ;
14 failat 11
15 run execute s:StartupII
16 wait >NIL: 5 mins; wait for StartupII to finish
17 ;
18 SYS:System/SetMap gb; Activate the ()/* on keypad
19 path ram: c: sys:utilities sys:system s: sys:prefs add
20 LoadWB delay; Wait for inhibit to end before continuing
21 endcli >NIL:
```

*Figure 1: Workbench 1.3 Startup-Sequence*

Line 6 causes any external devices that need special configuration to be initialised. *BindDrivers* looks in the Expansion drawer for special drivers, which means that all you need to do when installing a new piece of hardware is drag its icon into that drawer and reboot—very simple.

Lines 7 & 8 load the time and date from the battery backed-up clock, while line 9 speeds up text output by running a program called FastFonts. Line 10 installs the extended AmigaDos 1.3 shell (which is more powerful than the CLI). line 11 makes the Execute command resident—this means that it is permanently loaded into memory so whenever it is used, no disk access is required.

Line 12 installs a new device called NEWCON: which is basically a CLI window that allows editing with the cursor keys. Note that mounting a device doesn't immediately cause anything to happen, it only makes the device available to the system for later use.

Line 14 makes the script less easy to stop—if any of the commands after this fail for some reason, AmigaDOS will carry on executing the remaining commands so that Workbench gets loaded regardless.

Lines 15 and 16 work together to execute another script called StartupII, also in the S: directory. When StartupII is finished, it sends a signal back to the Startup-Sequence which causes the Wait command to abort, allowing the script to continue. You can look at StartupII yourself to see what it does.

Line 18 selects the UK keyboard layout (i.e. the one with the # over the 3). Line 19 sets a path for all CLI/Shell commands; this allows you to directly execute commands like Format and Clock that are stored in the System or Utilities directories, without having to type in a full pathname.

All that remains now is to load Workbench and remove the CLI window, which is neatly accomplished by lines 20 and 21. The `delay` parameter to `LoadWB` tells Workbench to wait until it has checked all the diskdrives for disks before it returns control to the Startup-Sequence. If this wasn't here, Workbench and the Startup-Sequence would both be trying to access the disk drives at the same time, and things would get very slow.

## So, how do I change it?

Now that you know what the Startup-Sequence does, let's go back to MEmacs and customise it a bit. Making sure you are using a BACKUP COPY of Workbench, let's get the Startup-Sequence to print your name on the screen each time you boot the system. Move the cursor to the beginning of line 5 (it reads `SYS:System/FastMemFirst`) and press Return to insert a blank line in the file. Now type

```
ECHO "Name's Workbench disk is booting......  *N"
```

substituting your name as appropriate. Now select *Save-Exit* from the *Project menu.* You'll get dumped out of MEmacs. Now reset the Amiga and boot with your modified Workbench disk. After the computer prints "A500/A2000 Workbench disk," it should display the message you entered.

Let's try something a bit more complex—starting the Clock program automatically on bootup. Reload MEmacs and load in the Startup-Sequence again. Now move the cursor to the line reading `LoadWB delay` and press Return. In the blank line, type

```
RUN SYS:Utilities/Clock
```

and then select *Save-Exit.* Reboot and you'll find Clock doing its thing automatically. Much the same procedure can be repeated for any program on the Workbench Disk.

## Okay, but what about programs on other disks?

To run them, you'll need to copy the relevant program to your Workbench disk. But wait, the Workbench is full to the brim!

There are two ways of making more space on the Workbench. One is to use a disk cruncher like *PowerPacker*, the other is to delete some useless stuff from the disk. Either way, create enough space on the disk for whatever programs you want copied over. By the way, don't delete anything from the L, Devs, C, Libs or S directories unless you know what you're doing.

The simplest way to copy over new programs is to drag the icon of the program to the Workbench disk (preferably the Utilities drawer). You can also use a directory utility like *SID* (from Fish Disk 338) or *ClickDOS*. Then use MEmacs to edit the Startup-Sequence. Create a blank line (below the line which runs the clock program) and type

```
RUN SYS:Utilities/ProgramName
```

assuming the program is in the Utilities drawer of the disk.

## More startup tricks

You can customise the Startup-Sequence to do virtually anything. For example, if your Amiga has lots of memory (say 2 megs) you can set it up to work with the RAD: device. RAD: is a ram disk that remembers its contents after a reset; if you have Kickstart 1.3, you can also boot from RAD:. By modifying the entry for RAD: in the file DEVS:Mountlist, you can change how much memory RAD: uses up (change the *HighCyl* value). To make RAD: available, use the following command:

```
MOUNT RAD:
```

If you have set your *HighCyl* value to 79, then RAD: will occupy 880K, exactly the same as a floppy disk. This allows you to use the following handy trick:

```
DISKCOPY DF0:  TO RAD:
```

This will copy the contents of the disk in DF0: (normally Workbench) into RAD:. If you need to get rid of RAD: later on, you'll have to use the `RemRad` command and then reboot.

Here's another useful thing you can try. Let's say that you have loads and loads of fonts. You've put them all on a disk called "MyFonts", and you always put this disk in your external drive each time you boot *Deluxe Paint*. However, you can't easily access the fonts from within DeluxePaint. The solution is to just follow these instructions. First, use SID to copy the Assign command to the C directory on the *DeluxePaint* disk. Next, edit the Startup-Sequence on that disk and add the following line anywhere near the start:

```
ASSIGN FONTS: MyFonts:
```

Now boot *DeluxePaint* with the fonts disk in your external drive and the extra fonts will be available from within *DeluxePaint*. A similar trick can be repeated for many other paint/animation/titling programs.

I would strongly advise you to put a virus killer in your Startup-Sequence. This gives you a warning if your WorkBench becomes infected.

When dealing with Startup-Sequences, the important thing to remember is to experiment. Don't be afraid to try a new program in your Startup-Sequence, although it is advisable to keep a backup of your custom Workbench which you've been working on for the past year and is brilliant, etc, in case it gets corrupted—it happened to me! Until next time...

---

## For your Diary

Here's a list of the meetings taking place over the next few months. Note that at Christmas, we are squeezing in an extra meeting for the quiz, and regular meetings will follow after that at the normal fortnightly intervals.

| | |
|---|---|
| 18 Oct 91 | Autumn Quiz |
| 1 Nov 91 | A.G.M.—be there! |
| 15 Nov 91 | Normal meeting |
| 29 Nov 91 | Yet another normal meeting |
| 13 Dec 91 | If you're superstitous, don't come |
| 20 Dec 91 | Christmas Quiz |
| 3 Jan 92 | First meeting of the New Year |
| 17 Jan 92 | Last meeting of January |

Please note that the items advertised may be subject to change without notice. See you at the A.G.M.

# Data Structures in C, Part 2

*by Kevin McDonnell*

In the last article, we looked at arrays of integers and chars. This time we're going to look at something slightly more complicated: structures and arrays of structures.

## Structures

A structure is a collection of one or more variables, grouped together to enable us to treat them as a single item. The thing to note is that these variables do not have to be of the same type. An example of a structure would be a record of club members. It might look something like this:

```
struct club_member {
    char  name[50];
    char  addr1[30];
    char  addr2[30];
    char  addr3[30];
    char  addr4[40];
    short member_type;
    short year_joined;
};
```

Note: Each item in the structure, e.g. `name`, is called a *member* of the structure. To avoid confusion for the rest of the article, *member* will refer to a member of a structure, i.e. a variable.

This gives us the structure `club_member` which holds the information we need about the member: name, address, type of membership and the year the member joined. Now that we have defined the structure, we need to declare an instance of it to use it in our programs. We do this as follows:

```
struct club_member member_info;
```

This declares a variable `member_info` which is of type `struct club_member`. To access the 'members'of the structure we use the *structure member operator* which is '.' (a single dot):

```
strcpy(member_info.name, "Joe Bloggs");
```

This copies the string "Joe Bloggs" into the field `name` of `member_info`. The program in figure 1 is an example of this; it reads in the values for each member from the keyboard, stores them in the structure and then displays them on the screen.

```
#include <stdio.h>
#include <stdlib.h>

struct club_member {
   char name[50];
   char addr1[30];
   char addr2[30];
   char addr3[30];
   char addr4[40];
   int  member_type;
   int  year_joined;
};

void main() {
   struct club_member mem;

   printf("Enter name :");              gets(mem.name);
   printf("Enter address [4 lines] :"); gets(mem.addr1);
   printf("                         "); gets(mem.addr2);
   printf("                         "); gets(mem.addr3);
   printf("                         "); gets(mem.addr4);
   printf("Enter member type :");
   scanf("%d", &mem.member_type);
   printf("Enter year joined :");
   scanf("%d", &mem.year_joined);

   disp_member_info(mem);
}

void disp_member_info(struct club_member cm)
{
   printf("Member's Name     : %s\n",cm.name);
   printf("Member's Address  : %s\n",cm.addr1);
   printf("                    %s\n",cm.addr2);
   printf("                    %s\n",cm.addr3);
   printf("                    %s\n",cm.addr4);
   printf("\n");
   printf("Membership Type   : %d\n",cm.member_type);
   printf("Year Joined       : %d\n",cm.year_joined);
}
```

*Figure 1: Program to read in structure members*

## Arrays of structures

Arrays of structures work in the same manner as the arrays we discussed last time. To declare an array of 10 `club_members`, we use:

```
struct club_member members_info[10];
```

We access the individual entries in the array like so:

```
members_info[0], members_info[1], .. ,members_info[9]
```

## Arrays and pointers

A pointer is a variable which contains the address of a variable. This sounds a bit weird but can be explained quite easily. Let's say we declare an integer variable `x` and it is placed in memory at address 20 by the computer. If we have a pointer to `x` it simply means that we have a variable which contains the address in memory that `x` is stored at, i.e. 20. The usefulness of this may not be readily apparent, but consider this declaration:

```
int x[10];
```

This declares an array of 10 integers, which if you remember from last time are held consecutively in memory. Now, if we have a pointer to `x`, we also by default have a pointer to the entire array. (Note that if the array name is used on its own, C will treat it as a pointer to the first element.) So how do we define a pointer? It's done like this:

```
int *ptr;
```

This declaration is interpreted as '`ptr` is a pointer to an integer,' i.e. `ptr` is a variable that contains the address of an integer variable. To get `ptr` to point to `x`, we use the statement:

```
ptr = x;            /* Ptr points to x[0] */
```

We can now use `ptr` to access the elements of `x`, like so:

```
*(ptr+1) = 0;       /* set x[1] = 0    */
ptr     += 5;       /* set ptr to x[5] */
*(ptr-4) = 4;       /* set x[1] = 4    */
```

An example program showing the use of pointers to address an array of integers is given in figure 2. This idea also extends to structures:

```
struct club_member memb_arr[20];
struct club_member *info_ptr;
```

declares an array of 20 `struct club_members` called `memb_arr` and a variable `info_ptr` which contains the address of a variable of type `struct club_member`. To get the pointer to point to the array we use:

```
info_ptr = &memb_arr[0];
```

```
#include <stdio.h>

void main()
{
   int i, x[10];
   int *ptr;

   for(i = 0; i < 10; i++)
     x[i] = i;

   ptr = &x[0];
   /*
    *   Print all numbers in descending
    *   order using the pointer set to x[0]
    *   and a counter i
    *
    *   NOTE: (ptr+i) is equivalent to x[i]
    */
   for(i = 9; i >= 0; i--)
     printf("%d\n", *(ptr+i));

   /*
    *   Print even numbers using the pointer only
    *
    *   NOTE: Directly incrementing the pointer
    *         changes the element pointed to
    *         accordingly.
    */
   for(ptr = &x[0]; ptr < &x[9]; ptr += 2)
     printf("%d\n", *ptr);
}
```

*Figure 2: Using pointers to address an array of integers*

and we can then use the pointer as before.

Well that's all for now. In the next article we'll talk more about pointers and we'll start to develop an application based on the `club_member` structure, a mini-database to hold the membership list for a small club. As usual if you have any questions forward them to me c/o CUGI and I'll do my best to answer them.